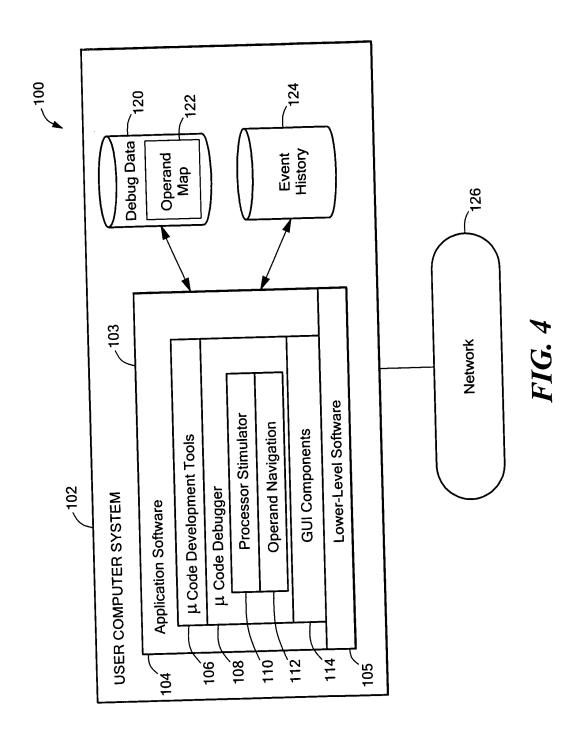
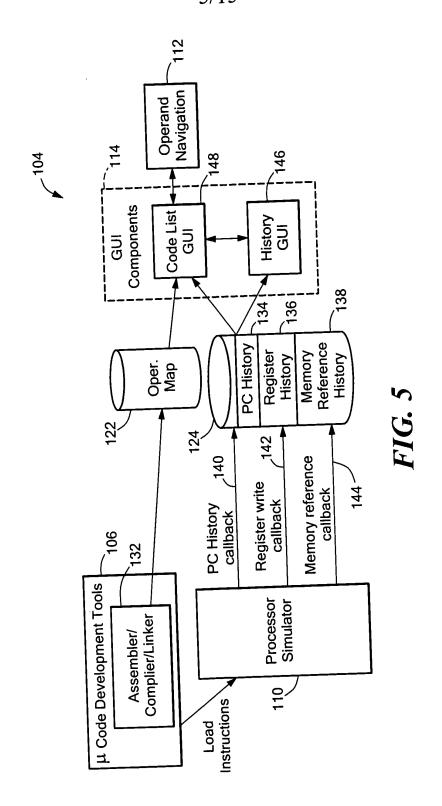


FIG. 3



5/15



Application No. 10/807,218

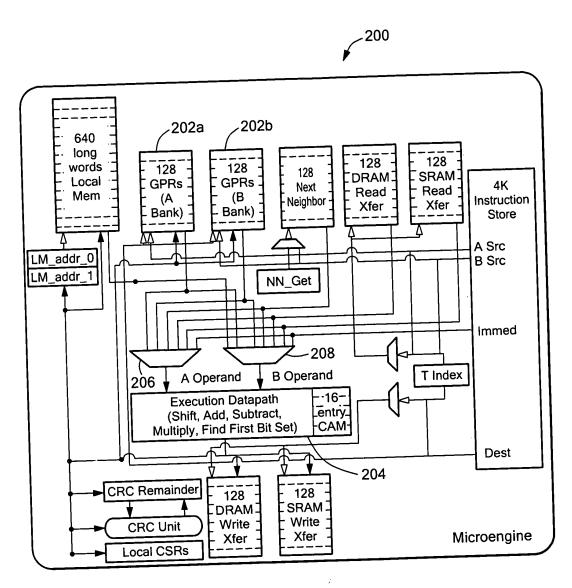
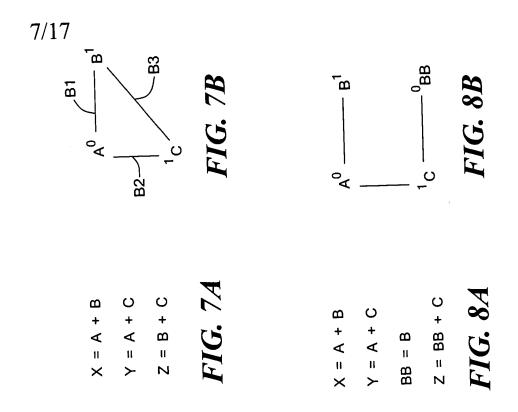
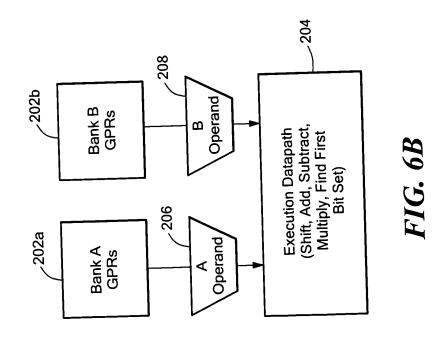


FIG. 6A

Application No. 10/807,218

7/15





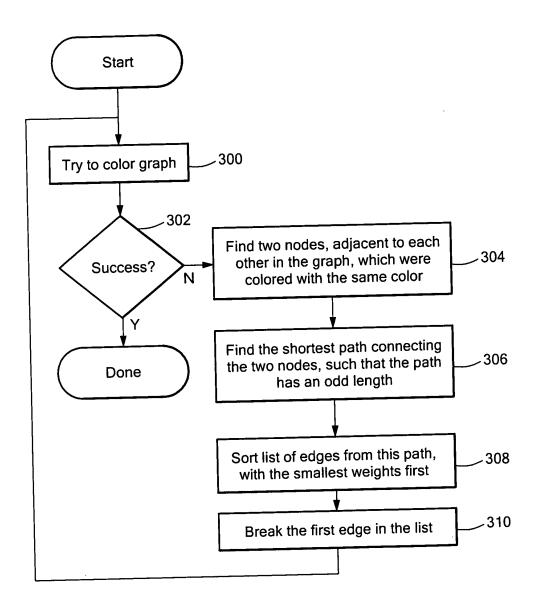


FIG. 9

James D. Guilford Application No. 10/807,218

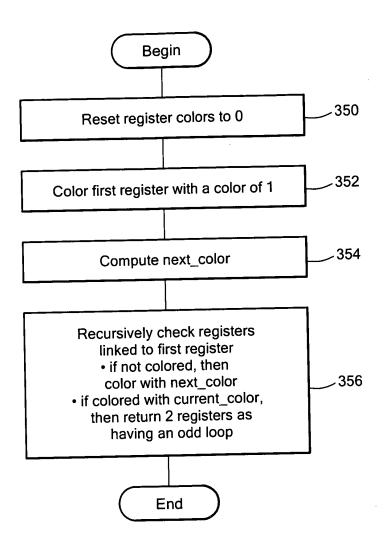


FIG. 10

DEBUG SYSTEM HAVING ASSEMBLER CORRECTING REGISTER ALLOCATION ERRORS James D. Guilford

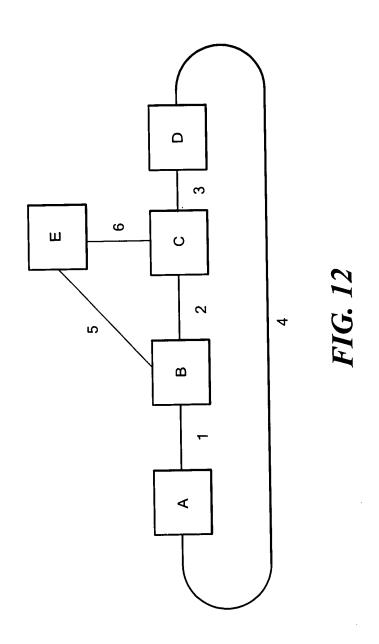
Application No. 10/807,218

10/15

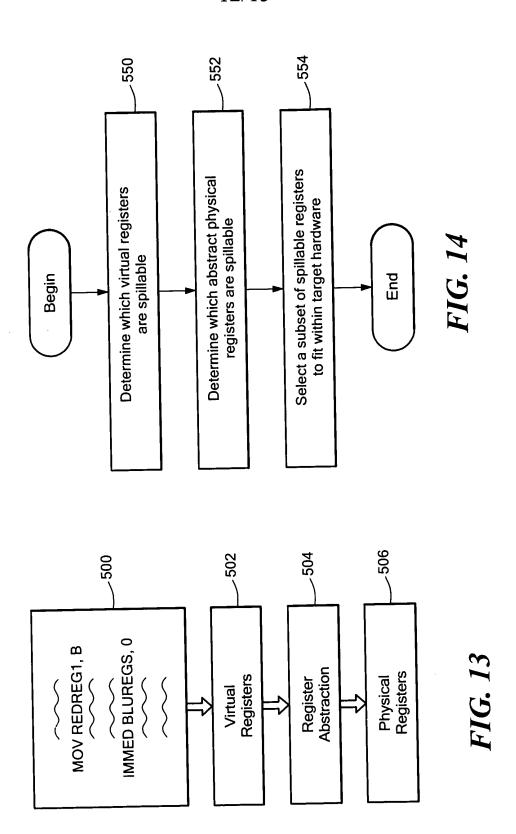
Begin The PATH pointers in all registers in the groups are set to NULL. 400 B's PATH[0]/PATH_WEIGHT[0] are set to point A. 402 Variable LIST is set to include only B, using LIST[0] pointers. 404 Variable DIST is set to 1 406 While LIST is not empty 408 Initialize NEW_LIST to empty 410 PREV_DIST is set to (DIST^1) 412 While REG in LIST (using LIST[PREV_DIST] pointers) 414 While LINK in list of edges from REG 416 If (LINK is broken) continue 418 If ((linked-reg==A)and(DIST ==1)), then even loop, continue to look for other loops 420 If ((linked-reg == A) and(DIST ==0)), then odd loop found: 422-Store the edge weight with linked-reg/A in PATH_WEIGHT [DIST] 424 Return REG 426-If (linked-reg->PATH[DIST] is empty) 428-Store REG and the edge weight with linked-reg in PATH/PATH_WEIGHT[DIST] 430-Push the linked-reg onto NEW_LIST 432 Set LIST to NEW LIST 434 Set DIST to (DIST^1) 436 End

FIG. 11

DEBUG SYSTEM HAVING ASSEMBLER CORRECTING REGISTER ALLOCATION ERRORS James D. Guilford Application No. 10/807,218



12/15



James D. Guilford Application No. 10/807,218

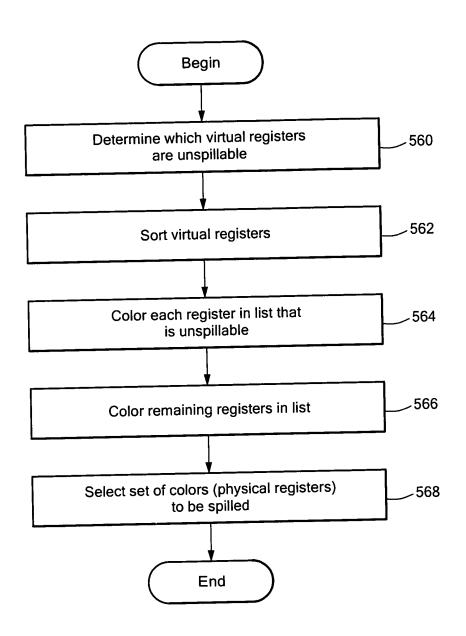
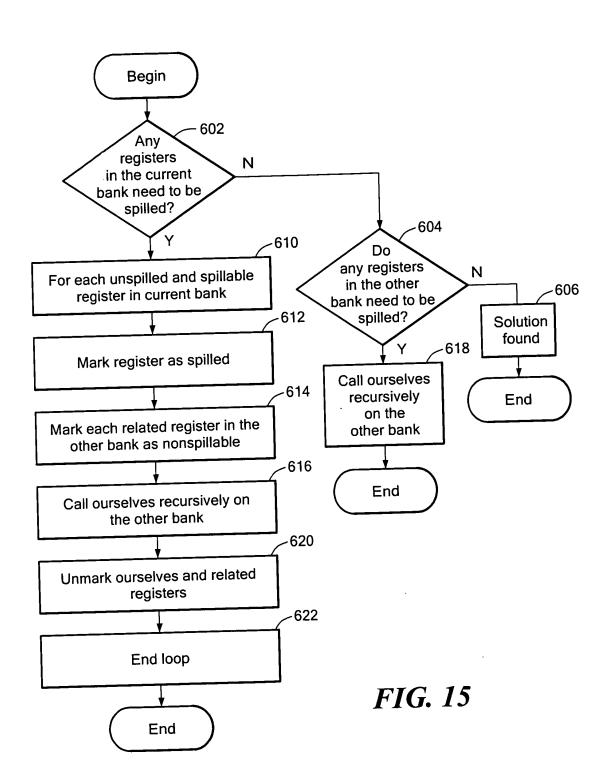


FIG. 14A

Application No. 10/807,218



James D. Guilford Application No. 10/807,218

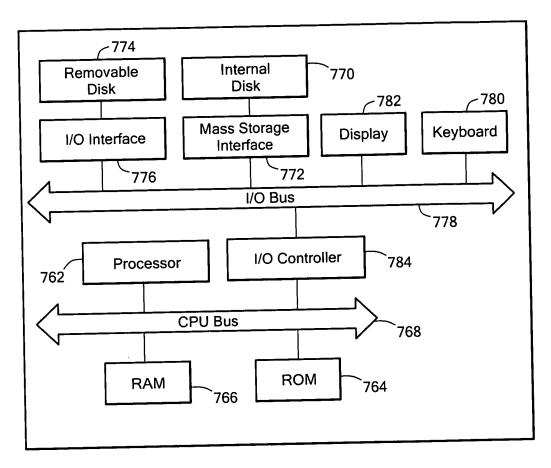


FIG. 16